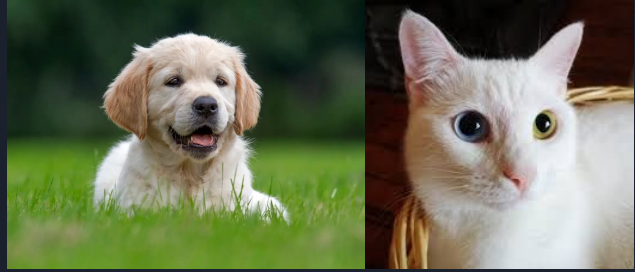# Intro to Machine Learning

Week 1 - Introduction

I have an idea!

Define the task you want computer to perform

- I want to teach computer to be able to distinguish cats and dogs.
- When a cat picture is given, computer should be able to correctly classify that animal as a cat.

How to do that

Start collecting images of dogs and cats.

Like humans, computers can also recognize the difference between cats and dogs.

To make human able to recognize between 2 species, they must observe some images in order to differentiate 2 species?

Similarly, for a machine, in order to capture the difference between two animals, it must observe many samples in order to classify accurately.

Design Features

While looking at a cat or a dog picture, where do people really focus? What are the key points that people focus on in order to say that it is a cat or a dog?

These features are important for computer in order to differentiate cats and dogs. Without guidance, learning would be hard and slow for them.

Let's say we choose 2 features as

1. Size of nose, relative to the size of the head
2. Shape of ears

We got features, now what?

We need to construct, find a function that able to do this;

F(cat image) = cat   F(dog image) = dog

We should select a model that should be able to learn in short amount of time, and predict accurately. For now the model is a black box that accepts an image of dog/cat and returns if the animal is cat or a dog.

Function F

Cat/Dog
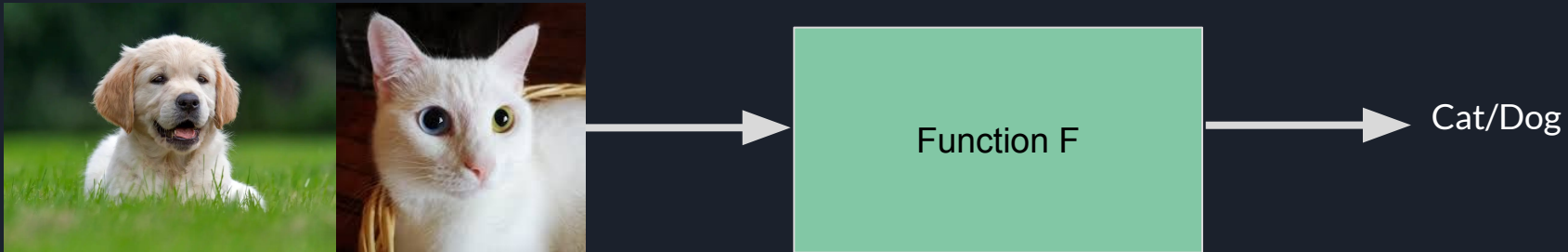
Examine it

It is time to put our function to test

Making our model to infer based on images that model did not see during training

Function F

?

Basic Pipeline of Machine Learning

1. Define a task
2. Collect Data
3. Design Features
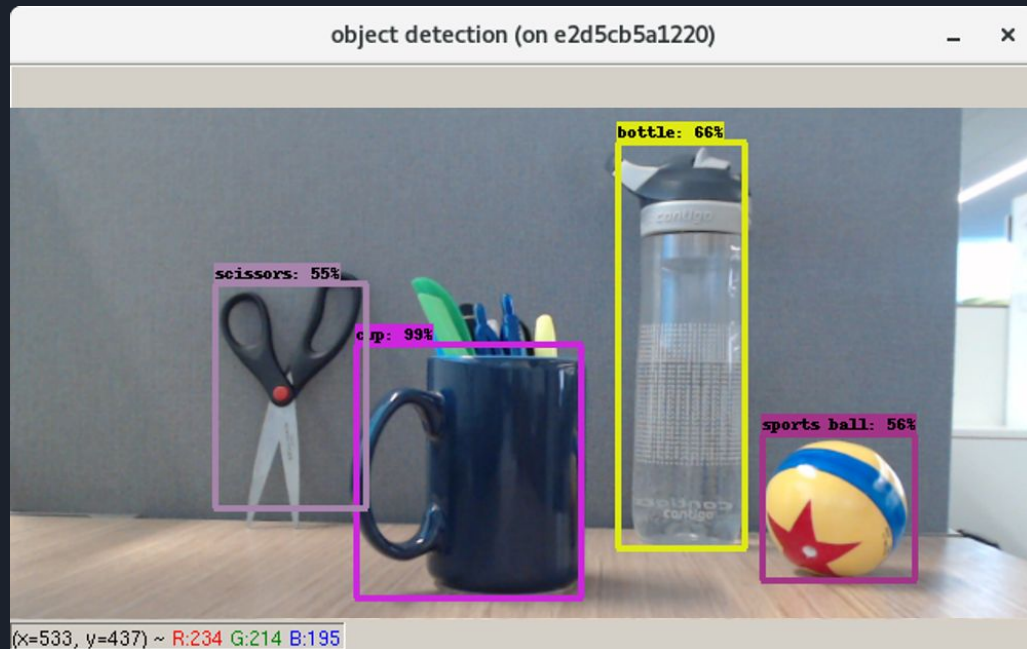4. Train Model
5. Test Model

2 Main Learning Problems

1.   Regression

To describe it shortly, suppose you are running a real estate company, and have 1000 houses. You want to predict a price for a new house. You have different features of the houses you process and want to use them for predicting the price for a new house.
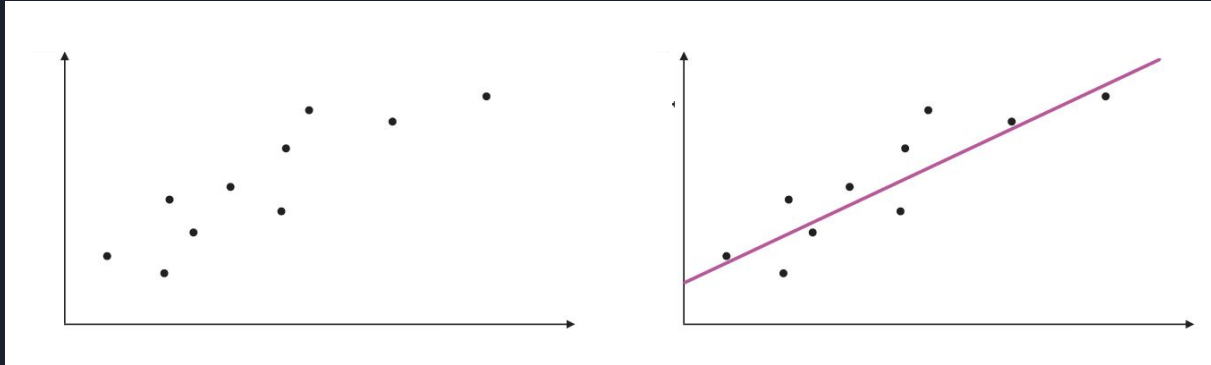
Example Use Case of Regression

Classification

As name suggests, classification tasks finds the line/hyperplane that divides 2 or more classes accurately. To give a basic example, cat dog recognition is a classification task which model needs to predict accurately whether the provided image is a cat or a dog.

Basic example would be a sentiment analysis. It classifies given sentence, whether it is a positive or a negative sentence.
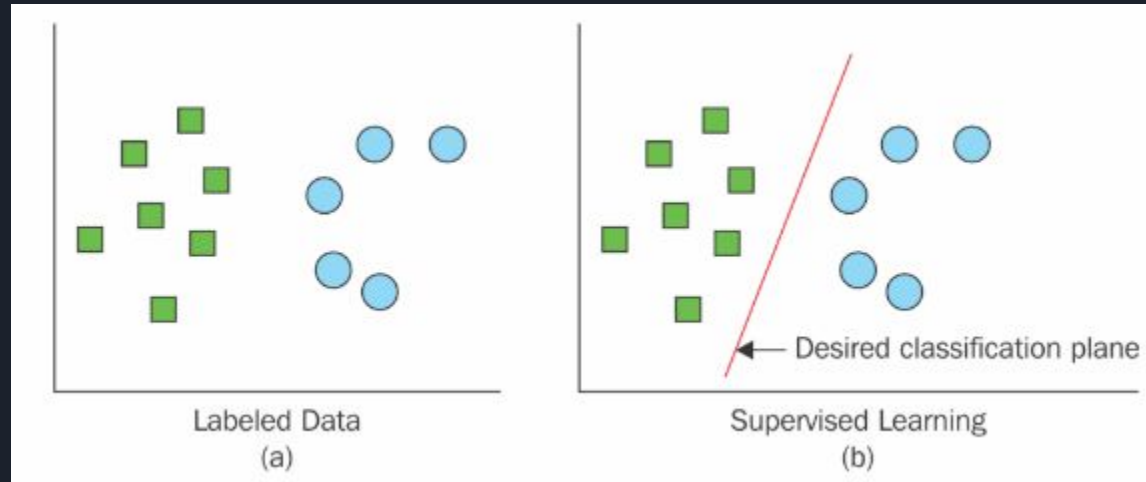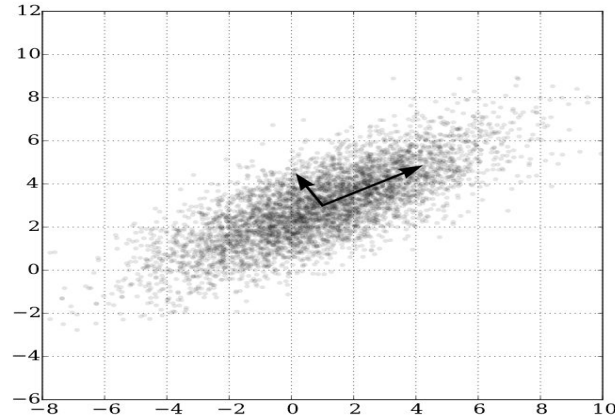
What if data is too complex

Hardware has its own limits that can handle training process.

There might data that will not affect the training process positively

We need to modify the data that is able to fit to its hardware and affect the training positively.

Optimization

We need to improve our function using different algorithms.

We need to find the parameters that will be able to perform the task we defined accurately.